

ICC
Projet de fin de module
WatchOurMovie (WOM)
Pour savoir quoi regarder...



Contexte de l'appliatif

Contexte :

« Il est parfois long de choisir quoi regarder le soir »

« Je passe 30 min à scroller Netflix sans jamais sélectionner quoi que ce soit »

« Nous n'arrivons jamais à nous mettre d'accord sur quoi regarder, c'est souvent source de dispute »



But de l'applcatif

Considérons un groupe de N personnes.

« Recommander, un ou plusieurs films, à un instant T , qui n'ont pas encore été vus par ce groupe de N personnes et qui partage un maximum de points commun entre ces personnes. »



Discussion autour de l'architecture

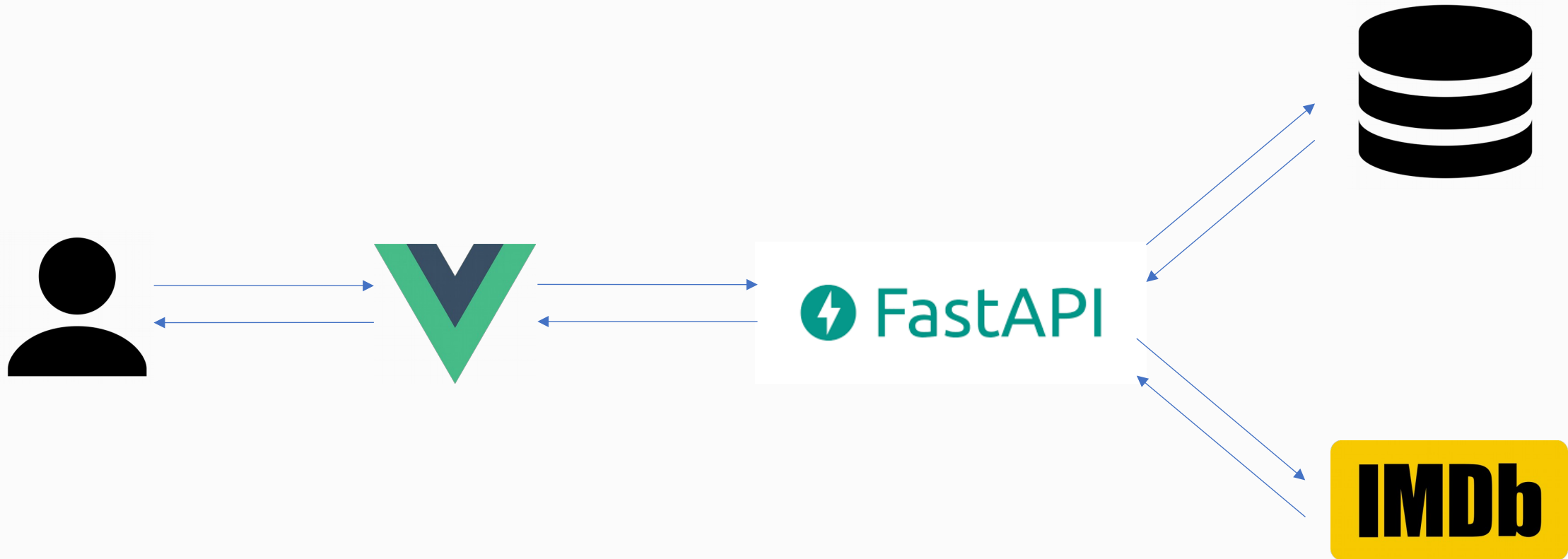
Comment résoudre cette problématique ?

Proposez vos solutions et vos architectures !



Ma proposition d'architecture

(soumise à modification selon les envies/choix des étudiants)



FrontEnd

- **CGU.**
- **Contacts.**
- **Présentation du site.**
- **Création de compte / Authentification.**
 - **Recherche de films.**
 - **Accès à des pages de présentation de film.**
 - **Labellisation de films (Vu / Non Vu).**
 - **Gestion de la préférence utilisateur (Genres de films préférés, langue de l'utilisateur).**
 - **Gestion de groupes entre utilisateurs (CRUD).**
 - **Génération de la recommandation pour un groupe donné.**



BackEnd

- **Gère l'authentification et les token d'authentification (JWT).**
- **Gère les groupes d'utilisateurs.**
- **Gère les préférences utilisateurs.**
- **Gère les films vu et non vus de l'utilisateur.**
- **Interagit avec l'API IMDB pour récupérer la donnée de films**
 - **Si la donnée a déjà été récupérée, la stocker en base pour éviter un appel à l'API IMDB (coûteux).**
 - **Sinon appeler l'API IMDB.**



Database

- **Devra stocker des logs applicatifs**
- **Gère les données utilisées par le Backend**
 - **Si les données ne sont pas présentes en base, appeler l'API IMDB**
 - **Sinon, envoyer ces données au BackEnd**



API IMDB

- Devra être utilisée pour récupérer la donnée initiale de films
- Devra être utilisée pour récupérer la donnée méconnue du système



Choix techniques

L'application devra utiliser :

- **Un backend FastAPI**
- **Un frontend VueJS**
- **Un framework CSS**
- **L'API IMDB <https://imdb-api.com/>**
- **Une base de données (MongoDB)**



Contraintes techniques

- **L'API IMDB limite le nombre de requête quotidien à 100. (Pensez à n'utiliser l'API que pour les requêtes dont vous ne connaissez pas encore l'information de retour.)**
- **Chaque brique applicative sera conteneurisée sous Docker.**
- **Le déploiement sera orchestré par Docker Compose.**
- **La base de données devra stocker des logs applicatifs pertinents.**



Petits tips de développement

- **Pensez à sortir toute constante qui pourra créer de la configuration dans un fichier séparé**
- **L'API est gratuite mais le nombre de requêtes par jour est limité. Mockez votre API avec votre propre donnée de test pour éviter de ne plus pouvoir faire d'appel. (<https://github.com/typicode/json-server>)**
- **Mettez vous à la place de l'utilisateur**
- **Pensez à bien faire vos schémas de base de données avant de démarrer le développement pour limiter les modifications de tables**



Notation

- **1 jour de retard = 4 points de moins**
- **Une application maintenable mais partiellement fonctionnelle sera plus valorisée qu'une application non maintenable mais entièrement fonctionnelle**
- **Réalisable en groupe de 3 personnes (1 groupe de 4 possible)**

Jalon de rendu : Dimanche 27 mars 2022 23h59

